

(19) World Intellectual Property Organization
International Bureau(43) International Publication Date
27 June 2002 (27.06.2002)

PCT

(10) International Publication Number
WO 02/50682 A2(51) International Patent Classification⁷: G06F 12/00

Avenue, #J207, Fremont, CA 94538 (US). WU, Chia, Y.; 37171 Sycamore Street, #727, Newark, CA 94560 (US). CHONG, Jr., Fay; 10019 Oakleaf Place, Cupertino, CA 95014 (US).

(21) International Application Number: PCT/US01/47891

(74) Agent: KOWERT, Robert, C.; Conley, Rose & Tayon, P.C., P.O. Box 398, Austin, TX 78767-0398 (US).

(22) International Filing Date: 11 December 2001 (11.12.2001)

(81) Designated States (national): AE, AG, AL, AM, AT, AU, AZ, BA, BB, BG, BR, BY, BZ, CA, CH, CN, CO, CR, CU, CZ, DE, DK, DM, DZ, EC, EE, ES, FI, GB, GD, GE, GH, GM, HR, HU, ID, IL, IN, IS, JP, KE, KG, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, MA, MD, MG, MK, MN, MW, MX, MZ, NO, NZ, PH, PL, PT, RO, RU, SD, SE, SG, SI, SK, SL, TJ, TM, TR, TT, TZ, UA, UG, UZ, VN, YU, ZA, ZW.

(25) Filing Language: English

(82) Designated States (regional): ARIPO patent (GH, GM, KE, LS, MW, MZ, SD, SL, SZ, TZ, UG, ZM, ZW), Eurasian patent (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European patent (AT, BE, CH, CY, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE, TR), OAPI patent (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, ML, MR, NE, SN, TD, TG).

(26) Publication Language: English

(83) Designated States (regional): ARIPO patent (GH, GM, KE, LS, MW, MZ, SD, SL, SZ, TZ, UG, ZM, ZW), Eurasian patent (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European patent (AT, BE, CH, CY, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE, TR), OAPI patent (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, ML, MR, NE, SN, TD, TG).

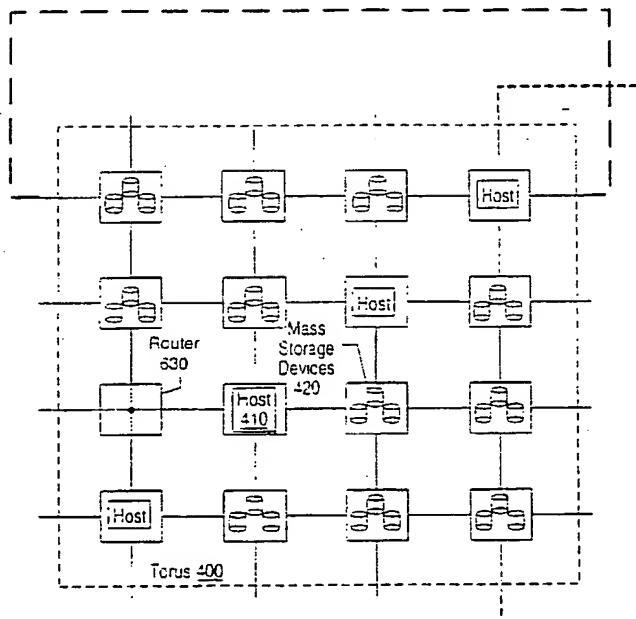
(30) Priority Data: 09/739,924 18 December 2000 (18.12.2000) US

(71) Applicant: SUN MICROSYSTEMS, INC. [US/US]; M/S UPAL 01-521, 901 San Antonio Road, Palo Alto, CA 94303 (US).

(72) Inventors: LEE, Whay, S.; 37171 Sycamore Street, Apartment 1438, Newark, CA 94560 (US). RETTBERG, Randall; 3308 Blackhawk Meadow Drive, Danville, CA 94506 (US). TALAGALA, Nisha, D.; 2000 Walnut

[Continued on next page]

(54) Title: HIGH PERFORMANCE STORAGE ARRAY INTERCONNECTION FABRIC USING MULTIPLE INDEPENDENT PATHS



(57) Abstract: A storage array interconnection fabric may be configured using multiple independent paths. A storage system including a plurality of communication paths is configured for connecting each node of a plurality of nodes forming an interconnection fabric. Each of the communications paths is an independent communications path. In addition, a first portion of the plurality of nodes is configured to communicate with a plurality of mass storage devices such as disk drives. A second portion of the plurality of nodes may be configured to communicate with a host.

WO 02/50682 A2

**Published:**

- without international search report and to be republished upon receipt of that report

For two-letter codes and other abbreviations, refer to the "Guidance Notes on Codes and Abbreviations" appearing at the beginning of each regular issue of the PCT Gazette.

**TITLE: HIGH PERFORMANCE STORAGE ARRAY INTERCONNECTION FABRIC USING
MULTIPLE INDEPENDENT PATHS**

BACKGROUND OF THE INVENTION

5

1. Field of the Invention

This invention relates to data storage systems and, more particularly, to storage array interconnection topologies.

10 **2. Description of the Related Art**

Computer systems are placing an ever-increasing demand on data storage systems. In many of the data storage systems in use today, data storage arrays are used. The interconnection solutions for many large storage arrays are based on bus architectures, such as small computer system interconnect (SCSI) or fibre channel (FC). In these architectures, multiple storage devices such as disks, may share a single set of wires, or a loop in the case of 15 FC, for data transfers.

Such architectures may be limited in terms of performance and fault tolerance. Since all the devices share a common set of wires, only one data transfer may take place at any given time, regardless of whether or not all the devices have data ready for transfer. Also, if a storage device fails, it may be possible for that device to render the remaining devices inaccessible by corrupting the bus. Additionally, in systems that use a single controller on each 20 bus, a controller failure may leave all the devices on its bus inaccessible.

There are several existing solutions available, which are briefly described below. One solution is to divide the devices into multiple subsets utilizing multiple independent buses for added performance. Another solution suggests connecting dual buses and controllers to each device to provide path fail-over capability, as in a dual loop FC architecture. An additional solution may have multiple controllers connected to each bus, thus providing a 25 controller fail-over mechanism.

In a large storage array, component failures may be expected to be fairly frequent. Because of the higher number of components in a system, the probability that a component will fail at any given time is higher, and accordingly, the mean time between failures (MTBF) for the system is lower. However, the above conventional solutions may not be adequate for such a system. To illustrate, in the first solution described above, the independent 30 buses may ease the bandwidth constraint to some degree, but the devices on each bus may still be vulnerable to a single controller failure or a bus failure. In the second solution, a single malfunctioning device may still potentially render all of the buses connected to it, and possibly the rest of the system, inaccessible. This same failure mechanism may also affect the third solution, since the presence of two controllers does not prevent the case where a single device failure may force the bus to some random state.

35

SUMMARY

Various embodiments of a high performance storage array interconnection fabric using multiple independent paths are disclosed. In one embodiment, a storage system including a plurality of communication paths configured for connecting each node of a plurality of nodes forming an interconnection fabric is disclosed. Each of 40 the communications paths is an independent communications path. In addition, a first portion of the plurality of

nodes is configured to communicate with a plurality of mass storage devices such as disk drives. In other embodiments, the mass storage devices may be random access memories configured as cache memories or tape drives. A second portion of the plurality of nodes may be configured to communicate with a host.

5 In some embodiments, each node of the plurality of nodes may be configured to communicate with each other node of the plurality of nodes by routing messages bi-directionally. In an alternative embodiment, each node of the plurality of nodes is configured to communicate with each other node of the plurality of nodes by routing messages uni-directionally.

10 In another embodiment, a method of interconnecting a plurality of nodes is recited. In one embodiment, each node is connected to each other node using a plurality of communications paths. The communications paths and the nodes form an interconnection fabric. Each of the communications paths is an independent communications path. Additionally, a first portion of the plurality of nodes is configured to communicate with a plurality of mass storage devices.

15 In an embodiment, a method for routing communications within a storage system comprising a plurality of nodes interconnected by an interconnection fabric is recited. In one embodiment, a communication from a source node is sent to a destination node using a first communication path. A failure in the first communication path may be detected. The communication from the source node may be resent to the destination node using a second communication path, which is independent from the first communication path.

BRIEF DESCRIPTION OF THE DRAWINGS

20 FIG. 1 is a diagram of one embodiment of an interconnection fabric using multiple independent paths;
FIG. 2 is a block diagram a node of an interconnection fabric, according to one embodiment;
FIG. 3A is a diagram of one embodiment of a torus interconnection fabric;
FIG. 3B is a diagram of one embodiment of a node configuration of a torus interconnection topology;
FIG. 4 is a diagram illustrating multiple independent paths between nodes in a system having a plurality of
25 nodes connected by a multiple independent path interconnection fabric, according to an embodiment;
FIG. 5 is a flow diagram of a method for routing communications between nodes in a multiple independent
interconnection fabric, according to an embodiment;
FIG. 6 is a flow diagram of another method for routing communications between nodes in a multiple
independent interconnection fabric, according to an embodiment;
30 FIG. 7 is a flow diagram of another method for routing communications between nodes in a multiple
independent interconnection fabric, according to an embodiment;
FIG. 8A is a diagram of one embodiment of a hypercube interconnection fabric;
FIG. 8B is a diagram of another embodiment of a hypercube interconnection fabric;
FIG. 9 is a diagram of an embodiment of a multiple path butterfly interconnection fabric;
35 FIG. 10 is a diagram of one embodiment of a complete graph interconnection fabric;
FIG. 11 is a diagram of one embodiment of a hex network interconnection fabric; and
FIG. 12 is a diagram of one embodiment of a fat tree interconnection fabric.

40 While the invention is susceptible to various modifications and alternative forms, specific embodiments thereof are shown by way of example in the drawings and will herein be described in detail. It should be understood, however, that the drawings and detailed description thereto are not intended to limit the invention to the

particular form disclosed, but on the contrary, the intention is to cover all modifications, equivalents and alternatives falling within the spirit and scope of the present invention as defined by the appended claims.

DETAILED DESCRIPTION OF EMBODIMENTS

5 Turning now to FIG. 1, a diagram of one embodiment of an interconnection fabric using multiple independent paths is shown. An interconnection fabric 100 is shown with several nodes. Each node may support one or more different types of devices in a storage system. The nodes are labeled with the letters C, H, M, R and S. A node with the letter C means the node may be configured to support a controller such as a Redundant Array of Inexpensive Disks (RAID) controller. A node with the letter H means the node may be configured with a host 10 interface or line card that may serve as an interface to a host computer. A node with the letter R means the node may be configured as a routing node and merely expands the communication paths available between other nodes. A node with the letter S means the node may be configured as a mass storage node and may be connected to one or 15 more mass storage devices, such as hard disk drives. A node with the letter M means the node may be configured as a storage cache memory node that provides, for example, a hierarchical storage cache for one or more mass storage nodes. Also, nodes may support any combination of these features. It is noted that while the nodes are configured and labeled in the embodiment of FIG. 1, this is only an exemplary drawing. In other embodiments, there may be other configurations that have a fewer or greater number of nodes and the nodes may be configured and used differently. For example, there may be a fewer or greater number of S nodes and a fewer or greater number of H nodes.

20 Generally speaking, each node may be connected to each other node in the fabric by multiple communication paths (not shown in Fig. 1). The communication paths form the fabric such that each communication path may be completely independent of each other path. Therefore, each node may have multiple possible paths to use when communicating with another node. Multiple independent paths may allow a source node and a destination node to continue communicating with each other even if one or more communications paths or 25 nodes between the source and destination nodes becomes inoperative. The interconnect fabric may be a point-to-point interconnect between each node, in which multiple independent paths exist between a source node and a destination node. In one embodiment, every node has multiple independent paths to communicate with every other node. The path independence of the fabric may allow a node or a path to fail or experience adverse conditions (e.g. congestion) without affecting any other node or path.

30 The figures that follow will describe an embodiment of a node of interconnection fabric 100 and some exemplary diagrams of possible forms that interconnection fabric 100 may take.

35 Turning now to FIG. 2, a block diagram of one embodiment of a node of the interconnection fabric of FIG. 1 is shown. In FIG. 2, a node 200 includes a routing unit 205 coupled to an interface controller 210. Routing unit 205 may be configured to communicate through multiple ports. In one particular embodiment, there may be four ports and the ports may be bi-directional. Thus, routing unit 205 may communicate with four neighboring nodes allowing four independent routing paths. In one alternative embodiment, routing unit 205 may be configured with four uni-directional ports: two inputs and two outputs. The choice between using bi-directional and uni-directional ports may be influenced by competing factors. The unidirectional design may be simpler, but it may only tolerate a single failure of a neighboring node. The bi-directional design tolerates more failures but may require a more complex routing unit 205. The size of the storage system array may be a determining factor, since for a very large 40 complex routing unit 205.

number of storage devices, a three-fault tolerant bi-directional fabric may become desirable to attain a reasonably low MTBF.

In addition to the nodes communicating with other nodes, in one embodiment, interface controller 210 may be configured to communicate with one or more disk drives 220. In another embodiment, interface controller 210 may be configured to communicate with one or more random access memories 230, such as a hierarchical storage cache memory or other type of memory and a memory controller. In yet another embodiment, interface controller 210 may be configured to communicate with a host or a RAID controller through a communication port, such as a peripheral component interface (PCI) bus. It is also contemplated that interface controller 210 may have all of these functions or any combination of the above described functions. For example, interface controller 210 may be 10 configurable for selecting between any one of the different types of interfaces described above. Thus, the ability to communicate with and/or control storage devices and communicate to hosts in an interconnection fabric may advantageously increase the reliability, performance and flexibility of large storage systems.

It is further contemplated that interface controller 210 may not have any devices attached. In such an embodiment, node 200 may simply connect to neighbors through routing port 205. Thus, node 200 may be used in 15 the interconnection fabric of FIG. 1 to increase the number of possible communications paths available. Therefore, some nodes may be unpopulated with storage or other devices, and used as a routing node to increase the number of paths in the interconnection fabric. Although it is contemplated that the above described node embodiments may be used in the following figures when nodes are discussed, there may be other embodiments of the nodes which are modifications of the above described node embodiments.

20 Referring to FIG. 3A, a diagram of one embodiment of a torus interconnection fabric is shown. A torus fabric 400 may be employed as the interconnection fabric depicted in FIG. 1. In FIG. 3A, torus fabric 400 uses a two-dimensional (2-D) array topology with the beginning nodes of each row and column connected to the respective endpoints of each row and column. For example, if the 2-D array is an N by M array, where N and M are both positive integers, then the first node in row one would be connected to the last node in row one, in addition to all the other nodes neighboring the first node. Likewise, from a column perspective, the top node in column one is connected to the bottom node in column one in addition to all the other nodes neighboring the top node. The remaining nodes are connected in similar fashion such that every node in the fabric of torus 400 is connected to its 25 four neighboring four nodes. It is noted that torus 400 is shown as a flat two-dimensional array with longer connections between the endpoints. These may be logical connections, and the physical layout of the nodes may be different. For example, each row may be physically oriented in the shape of a ring, such that the distance from the last node to the first node may be nearly the same as the distance between all the other nodes and likewise for the 30 columns.

The level of interconnection described above for a torus interconnect fabric means that in one embodiment each node may have four ports with which to communicate to the other nodes. In one embodiment, each of the four 35 ports is a bi-directional port, thus allowing both inputs and outputs from each neighbor. In an alternative embodiment, each of the four ports is a uni-directional port, thus allowing two inputs and two outputs. Thus, torus 400 may provide an interconnection fabric with multiple independent paths for a storage device system.

Although the above torus 400 is described using a two-dimensional array, it is contemplated that this same fabric may be extended to include a multi-dimensional array beyond two dimensions (not shown). One embodiment

of a three dimensional array may include several two-dimensional arrays "stacked" or layered such that each node now has six neighboring nodes instead of four and each layer is connected together using the two additional ports.

In an additional embodiment, torus 400 may be reduced to a mesh (not shown). A mesh, like torus 400, may be logically arranged in either a 2D or 3D array. However a mesh does not have the wrap around connections that the torus has, which connect the row and column endpoints together. Although the mesh does have multiple independent paths with which the nodes may communicate, not all the nodes have the same number of multiple independent paths.

Referring now to FIG. 3B, a diagram of one embodiment of a node configuration of a torus interconnection topology is shown. The torus topology 400 of FIG. 3A is shown here with some of the interconnections not shown for clarity. In torus 400 of FIG. 3B, a portion of the nodes is shown comprising storage devices, such as storage devices 420. In one embodiment, storage devices 420 may be disk drives. Another portion of the nodes are shown with host blocks in them, such as host 410. Host 410 may be a host communication port or line card. Other nodes, such as router node 630, may include a routing unit to expand the interconnect paths but may not include a device such as a disk drive or host interface. Thus, a storage system may include a plurality of nodes connected together by an interconnect fabric such as a torus fabric. The interconnect fabric may provide multiple independent point-to-point communication paths between nodes sending communications and nodes receiving the communications. A portion of the node may include mass storage devices such as hard drives. Other nodes may include storage controllers or host interfaces. In general, a mass storage system may be provided by the plurality of nodes and interconnect paths. The multiple independent paths between nodes may provide fail-over redundancy and/or increased bandwidth for communications between source and destination nodes. As mentioned above, many large storage systems use a large number of disks. To reduce costs, inexpensive and smaller disks may be used. However, since more disks may increase the failure rate, a highly redundant interconnection fabric, such as torus 400 may be used to provide a reliable overall system. For example, a storage controller node may send a write command and write data to a storage node having one or more hard drives. If the first path chosen for the write command fails, the command may be resent on a second path.

Additionally, the multiple paths of the torus interconnect allow for multiple parallel communications and/or disk operations that may be initiated over different paths, thereby possibly increasing the bandwidth and performance of the storage system. In a torus storage system with multiple controllers/host attachments, many parallel paths may exist between the hosts and the disks. Thus, many disk operations may be issued at the same time, and many data transfers may take place concurrently over the independent paths. This concurrency may provide a performance advantage and more scalability over bus-based architectures in which multiple devices must take turns using the same wires/fibre.

It is noted that other embodiments may use fewer or more storage devices 420 and fewer or more host 410 nodes to facilitate cost and performance tradeoffs. In addition, and as mentioned above, it is contemplated that some nodes may be configured to communicate with RAID controllers, and/or storage cache memory.

The torus fabric is just one example of a multiple path independent interconnect that may provide improved reliability and performance, as described above. Other examples are described below.

Turning now to FIG. 4 a plurality of nodes connected by an interconnection fabric using multiple independent paths is illustrated. No particular interconnect fabric scheme is shown since various different multiple independent path interconnects may be employed. In one embodiment, the nodes are connected by a torus fabric.

FIG. 4 shows one possible combination of four independent paths from source node S to destination D. Many other combinations of such redundant paths are possible. Note that each path may traverse multiple intermediate nodes between the source and destination.

Referring now to FIG. 4 and FIG. 5 together, a method is illustrated for routing communications within a multiple independent path interconnect fabric. A new communication may begin at source node S, as indicated at 560. To communicate with destination node D, source node S may attempt to use path 1, as indicated at 562. If the attempt succeeds, the communication event is completed, as indicated at 564 and 582. The attempt may fail due to various conditions in the path, including a failure in an intermediate node, congestion etc. If the attempt fails, the source node S retries the communication through path 2, as indicated at 564 and 566. If that also fails, source node 10 S tries path 3, as indicated at 568 and 570, and if that fails too, path 4 may be tried, as indicated at 572 and 574. After all of the paths have been tried without success, the source node S may optionally decide to return to path 1 and repeat the entire procedure again, as indicated at 578. In one embodiment, if the failure persists after some number of such repeated attempts, the source node may declare the destination node unreachable, and fail the operation completely, as indicated at 580.

FIG. 6 shows another approach for routing communications within a multiple independent path interconnect fabric. A communication may begin at source node S, as indicated at 660. Instead of sequentially trying path 1 through 4 (e.g., as in FIG. 5), the source node S may choose randomly from the possible paths 1 through 4, as indicated at 662. Source node S may retry until the operation is successful, as indicated at 664 and 670, or until the threshold is exceeded, upon which the destination is declared unreachable, as indicated at 666 and 668. Other path selection algorithms are also contemplated, such as a scheme in which paths are chosen by the source node according to a weighted preference assigned to each independent path from the source node to the destination node.

In the embodiments described in regard to FIG. 5 and FIG. 6, the intermediate nodes (e.g. those making up the path from S to D) may not make any decisions regarding what paths to try. In some embodiments, the intermediate nodes do not have complete knowledge of the path. For example, an intermediate node may only know that some message or communication came in from one of its input ports, requesting to go out a specified one of its four output ports. The intermediate nodes may simply attempt to pass along the message or communication from the input port to the requested output port. If the attempt succeeds, the communication/message progresses to the next node, until the message reaches its destination, upon which the message is delivered to the target device. Otherwise, 30 the path may be considered bad or congested, etc. This condition may be signaled back to the source (e.g. with the cooperation of upstream intermediate nodes in the path). This path failure notification may prompt the source to select another path for the retry, e.g. according to the methods shown in FIG. 5 or FIG. 6, or other alternatives.

Turning now to FIG. 7, a method is illustrated for routing communications within an interconnect fabric between nodes in which intermediate nodes may choose alternate paths upon detection of failures or adverse routing conditions. As used herein, an adverse routing condition may be any of various conditions that may cause slow or unreliable communications. An example of an adverse routing condition may therefore be a particularly congested path or a path with transmission errors. As used herein, a failure may or may not be a hard failure. For example, a failure may be declared if a path has an adverse routing condition for an extended period.

A communication may be sent from a source node to a destination node on a first communication path as 40 indicated at 300. A failure may or may not be detected on the first communication path from the source node as

indicated at 302. If no failure is detected the communication continues on to the next node as indicated at 316. If a failure is detected the communication may be resent on a second communication path as indicated at 304. Since the interconnect fabric described above may provide multiple independent communication paths from each node, in one embodiment, this procedure may be repeated in case the second communication path and a third communication path fails as indicated at 306 through 314. If a fourth communication path fails then an error may be declared. Assuming that at least one path from the source node was working, the communication continues to the next node as indicated at 316. If the next node is a destination node then the routing process is complete as indicated at 318, otherwise the routing procedure may be repeated for the next node. Alternatively, in the event that a failure is detected at 302, 306 and 310, a message or signal may be sent back to the source node indicating the failure and the source node may then choose an alternate path. It is noted that while this embodiment describes four paths, other embodiments may have a fewer or greater number of independent paths between each node.

Turning now to FIG. 8A, a diagram of one embodiment of a hypercube interconnection fabric is shown. A hypercube 500 may be employed as the interconnection fabric depicted in FIG. 1. In FIG. 8A, hypercube 500 has 8 nodes. Each node in hypercube 500 is connected to 3 neighboring nodes by three independent communications paths. Similar to the interconnection fabric shown in FIG. 1 and the torus interconnection fabric of FIG. 3A and FIG. 3B, the nodes of hypercube 500 of FIG. 8A may also be configured to control or be connected to devices such as hard disks, cache memories, RAID controllers and host communications interfaces.

In general, a hypercube may be thought of as a structure with 2^n nodes. Hypercube 500 may be created, for example, by starting with a rectangle containing four nodes (e.g. a 2^2 hypercube). To expand the structure, the 4 nodes are duplicated and connected to the existing 4 nodes forming hypercube 500, which is a 2^3 hypercube. The nodes in the duplicated structure are connected to the nodes in the existing structure that are in the same location in the structure. Additionally, the value of the exponent 'n' may also identify the number of independent paths connected to each node.

Thus, if a node or communication path fails, another path may be used to communicate. For example, node A of FIG. 8A is communicating with node D via a communication path 510. In the event that communication path 510 is detected as a failing path, an alternate path may be used. For example, the communication may be rerouted through the path including communication path 511, node B, communication path 512, node C and communication path 513.

Referring to FIG. 8B, a diagram of another embodiment of a hypercube interconnection fabric is shown. A hypercube 550 may be employed as the interconnection fabric depicted in FIG. 1. In FIG. 8B, hypercube 550 has 16 nodes. Hypercube 550 is an example of a 2^4 hypercube. Each node in hypercube 550 is connected to 4 neighboring nodes by 4 independent communications paths. Thus hypercube 550 is also an interconnection fabric with multiple independent communication paths. Similar to the hypercube described in FIG. 8A, the nodes of hypercube 550 of FIG. 8B may also be configured to control or be connected to devices such as hard disks, cache memories, RAID controllers and host communications interfaces.

Hypercube 550 may be constructed by duplicating the 2^3 hypercube in FIG. 8A. Each node in the original structure is connected to each node in the duplicated structure that is in the same location in the hypercube. For example, node A in FIG. 8B is connected to node I and node B is connected to node J and so on for the remaining nodes.

Additionally, the multiple paths of hypercube 500 of FIG. 8A and hypercube 550 of FIG. 8B may allow for multiple parallel communications and/or disk operations that may be initiated over different paths, thereby possibly increasing the bandwidth and performance of the storage system. In a hypercube storage system with multiple controllers/host attachments, many parallel paths may exist between the hosts and the disks. Thus, many disk 5 operations may be issued at the same time, and many data transfers may take place concurrently over the independent paths. This concurrency may provide a performance advantage and more scalability over bus-based architectures in which multiple devices must take turns using the same wires/fibre.

Referring to FIG. 9, a diagram of an embodiment of a multiple path butterfly interconnection fabric is shown. A butterfly interconnection fabric 650 may be employed as the interconnection fabric depicted in FIG. 1. 10 Butterfly interconnection fabric 650 includes nodes 610 and switches 620, which are interconnected via multiple communications paths. Similar to the interconnection fabric shown in FIG. 1 and the torus interconnection fabric of FIG. 3A and FIG. 3B and the hypercubes of FIG. 8A and FIG. 8B, nodes 610 and switches 620 of butterfly fabric 650 may communicate over multiple independent paths. Likewise, the nodes of butterfly 650 of FIG. 9 may also be configured to control or be connected to devices such as hard disks, cache memories, RAID controllers and host 15 communications interfaces.

Butterfly interconnection fabric 650 may be referred to as a 2-path 8-node butterfly. In other embodiments, butterfly interconnection fabric 650 may be expanded into a Benes network (not shown), which is two back-to-back butterflies.

Additionally, the multiple paths of butterfly 650 of FIG. 9 may allow for multiple parallel communications 20 and/or disk operations that may be initiated over different paths, thereby possibly increasing the bandwidth and performance of the storage system. In a butterfly storage system with multiple controllers/host attachments, many parallel paths may exist between the hosts and the disks. Thus, many disk operations may be issued at the same time, and many data transfers may take place concurrently over the independent paths. This concurrency may provide a performance advantage and more scalability over bus-based architectures in which multiple devices must 25 take turns using the same wires/fibre.

Turning to FIG. 10, a diagram of one embodiment of a complete graph interconnection fabric is shown. A complete graph interconnection fabric 700 may be employed as the interconnection fabric depicted in FIG. 1. In FIG. 10, complete graph interconnection fabric 700 includes nodes coupled together by multiple independent 30 communications paths. Similar to the interconnection fabrics described in the above FIGs., the nodes of complete graph interconnection fabric 700 of FIG. 10 may also be configured to control or be connected to devices such as hard disks, cache memories, RAID controllers and host communications interfaces.

Referring to FIG. 11, a diagram of one embodiment of a hex network interconnection fabric is shown. A hex interconnection fabric 800 may be employed as the interconnection fabric depicted in FIG. 1. In FIG. 11, hex 35 interconnection fabric 800 includes nodes interconnected by multiple independent communications paths. Similar to the interconnection fabrics described in the above FIGs., the nodes of hex interconnection fabric 800 of FIG. 11 may also be configured to control or be connected to devices such as hard disks, cache memories, RAID controllers and host communications interfaces.

Turning now to FIG. 12, a diagram of one embodiment of a fat tree interconnection fabric is shown. A fat tree 40 interconnection fabric 900 may be employed as the interconnection fabric depicted in FIG. 1. The fat tree interconnection fabric 900 of FIG. 12 includes nodes interconnected by multiple independent communications paths.

Similar to the interconnection fabrics described in the above FIGs., the nodes of fat tree interconnection fabric 900 of FIG. 12 may also be configured to control or be connected to devices such as hard disks, cache memories, RAID controllers and host communications interfaces.

Additionally, the multiple paths of the interconnection fabrics described in FIG. 10, FIG. 11 and FIG. 12 5 may allow for multiple parallel communications and/or disk operations that may be initiated over different paths, thereby possibly increasing the bandwidth and performance of the storage system. In a storage system with multiple controllers/host attachments, such as those described above, many parallel paths may exist between the hosts and the disks. Thus, many disk operations may be issued at the same time, and many data transfers may take place concurrently over the independent paths. This concurrency may provide a performance advantage and more 10 scalability over bus-based architectures in which multiple devices must take turns using the same wires/fibre.

Numerous variations and modifications will become apparent to those skilled in the art once the above disclosure is fully appreciated. It is intended that the following claims be interpreted to embrace all such variations and modifications.

WHAT IS CLAIMED IS:

1. A storage system comprising:
a plurality of nodes;
5 one or more mass storage devices connected to each one of a first portion of said plurality of nodes respectively; and
an interconnection fabric configured for connecting each one of said plurality of nodes to every other one of said plurality of nodes;
wherein the interconnection fabric comprises a plurality of point-to-point connections between said plurality of nodes, wherein said interconnection fabric is configured to provide a plurality of independent
10 communication paths between each one of said plurality of nodes to every other one of said plurality of nodes.
2. The storage system as recited in claim 1, wherein each node is configured to communicate with each other node by routing messages bi-directionally.
15
3. The storage system as recited in claim 1, wherein said mass storage devices comprise disk drives.
4. The storage system as recited in claim 1, wherein said mass storage devices comprise random access memories configured as storage cache.
20
5. The storage system as recited in claim 1, wherein said mass storage devices comprise tape drives.
6. The storage system as recited in claim 1, wherein said mass storage devices comprise optical storage drives.
25
7. The storage system as recited in claim 1, wherein each node is configured to communicate with each other node by routing messages uni-directionally.
8. The storage system as recited in claim 1, wherein a second portion of said plurality of nodes is configured
30 to communicate with one or more host computers.
9. A method of interconnecting a plurality of nodes in a storage system, said method comprising:
connecting each node to each other node using a plurality of point-to-point connections;
forming an interconnection fabric comprising the nodes and said point-to-point connections;
35 a source node sending a first message to a destination node over a first communication path in said interconnection fabric;
said source node sending a second message to said destination node over a second communication path in said interconnection fabric, wherein said second communication path is independent from said first communication path;
40 said destination node interfacing to a mass storage device to respond to said first and second communications.

10. The method as recited in claim 9 further comprising nodes on said first and second communication path routing messages bi-directionally.

11. The method as recited in claim 9, wherein said mass storage devices comprise disk drives.

5

12. The method as recited in claim 9, wherein said mass storage devices comprise tape drives.

13. The method as recited in claim 9, wherein said mass storage devices comprise optical storage drives.

10 14. The method as recited in claim 9, wherein said storage devices comprise random access memories configured as cache memories for caching data stored in one or more mass storage devices.

15. The method as recited in claim 9 further comprising said source node interfacing to a host.

15 16. The method as recited in claim 9 further comprising nodes on said first and second communication path routing messages uni-directionally.

17. A method for routing communications within a storage system comprising a plurality of nodes interconnected by an interconnection fabric, the method comprising:

20 sending a communication from a source node to a destination node using a first communication path comprising one or more point-to-point connections between said source node, any intervening nodes, and said destination node;

detecting a failure in said first communication path; and

resending said communication from said source node to said destination node using a second communication path

25 which is independent from said first communication path, wherein said second communication path comprises one or more point-to-point connections between said source node, any intervening nodes, and said destination node.

18. The method as recited in claim 17, further comprising:

30 detecting a failure in said second communication path; and

resending said communication from said source node to said destination node using a third communication path which is independent from said first and said second communication paths, wherein said third communication path comprises one or more point-to-point connections between said source node, any intervening nodes, and said destination node.

35

19. The method as recited in claim 18, further comprising:

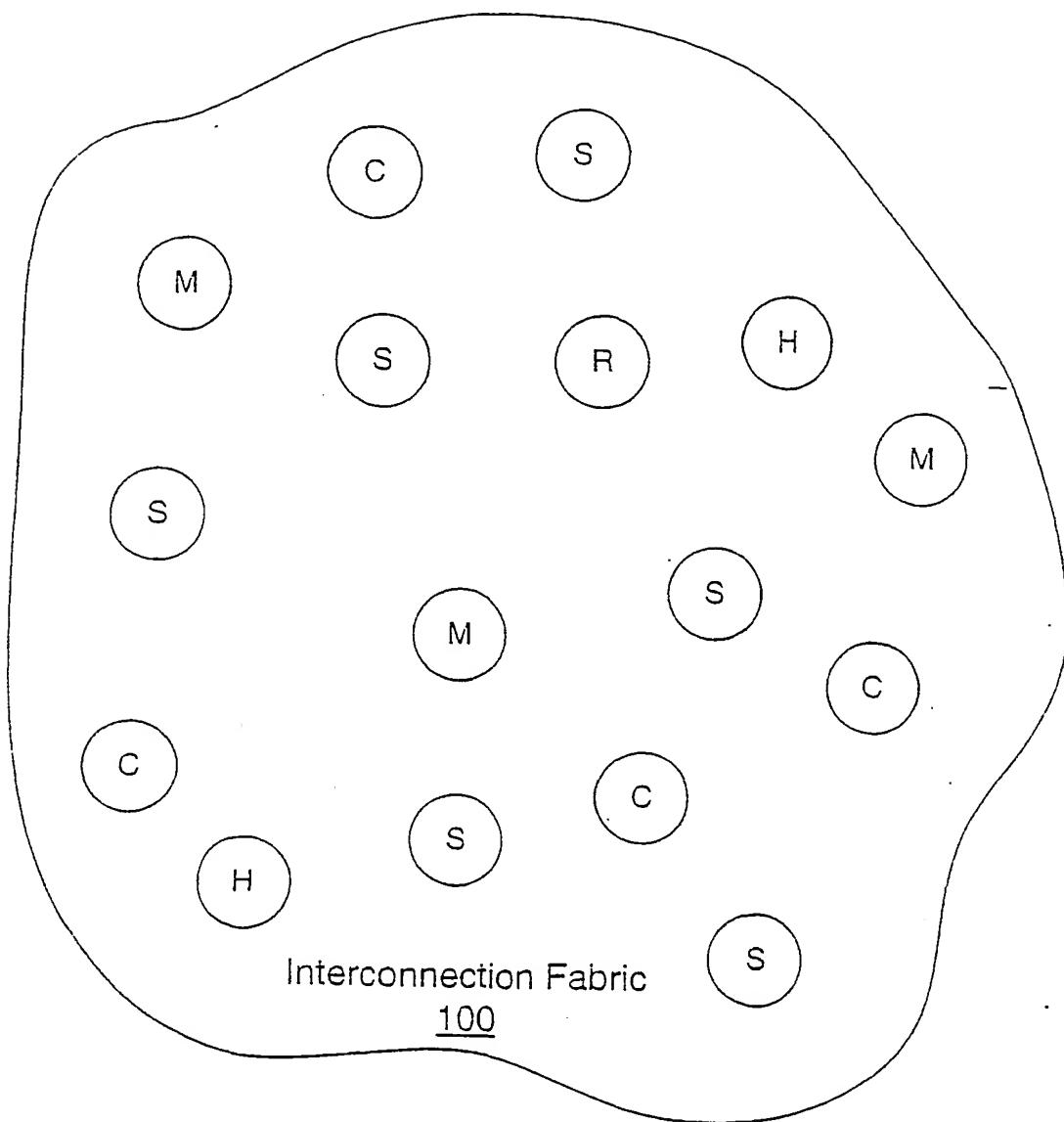
detecting a failure in said third communication path; and

resending said communication from said source node to said destination node using a fourth communication path which is independent from said first, said second and said third communication paths, wherein said fourth

communication path comprises one or more point-to-point connections between said source node, any intervening nodes, and said destination node.

20. The method as recited in claim 17 further comprising said destination node interfacing to a plurality of 5 mass storage devices.

21. The method as recited in claim 17 further comprising said source node interfacing to a plurality of mass storage devices.



H - Host Interface or Line Card
S - Mass Storage (e.g. Disk Drive)
M - Memory (e.g. Storage Cache)
C - Controller, (e.g. RAID Controller)
R - Router Node

FIG. 1

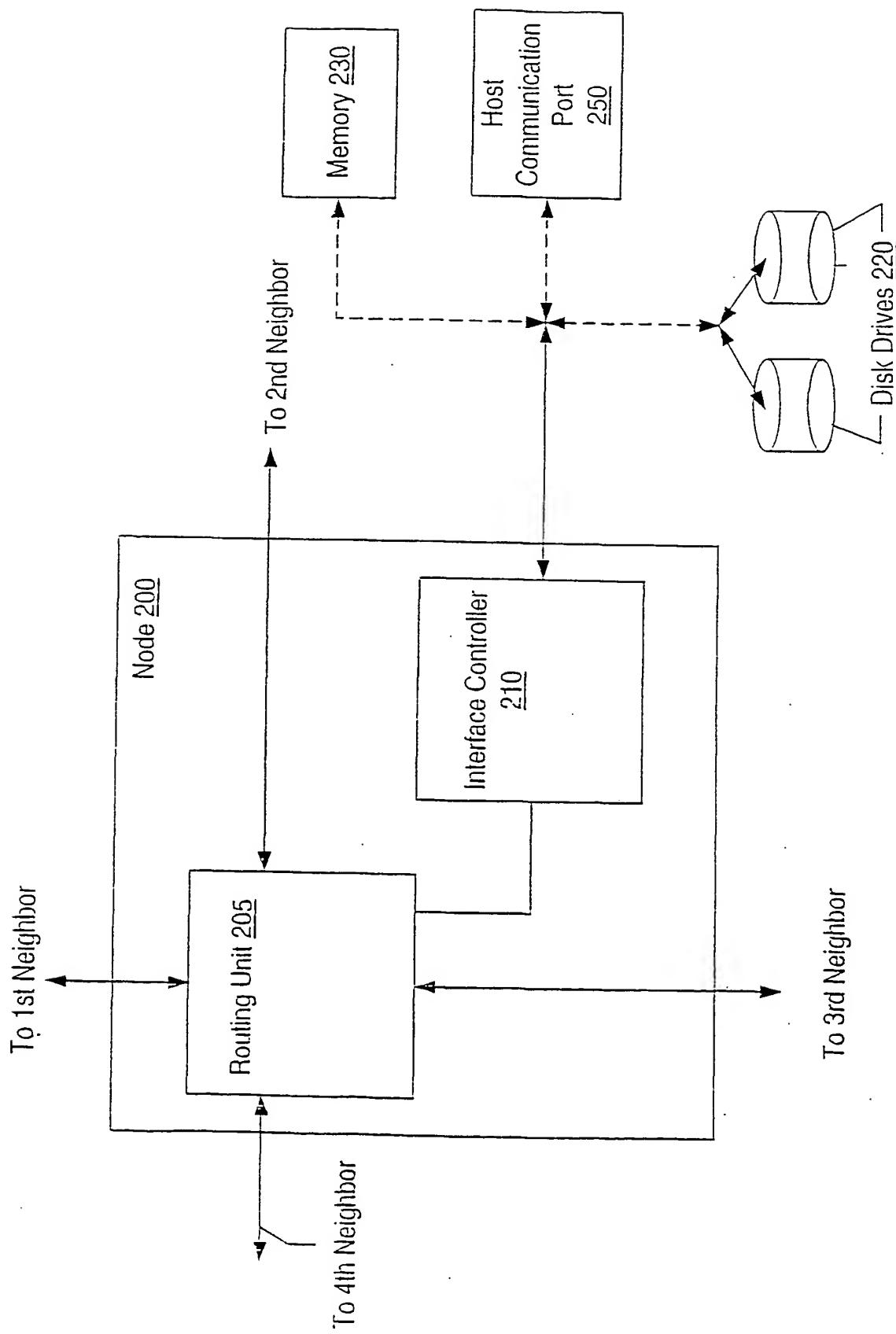
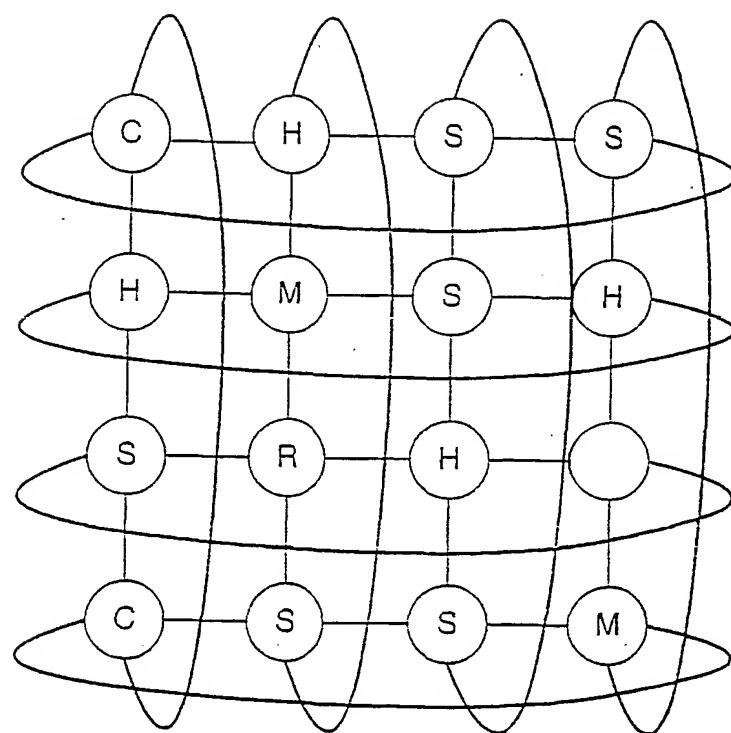


FIG. 2



400

FIG. 3A

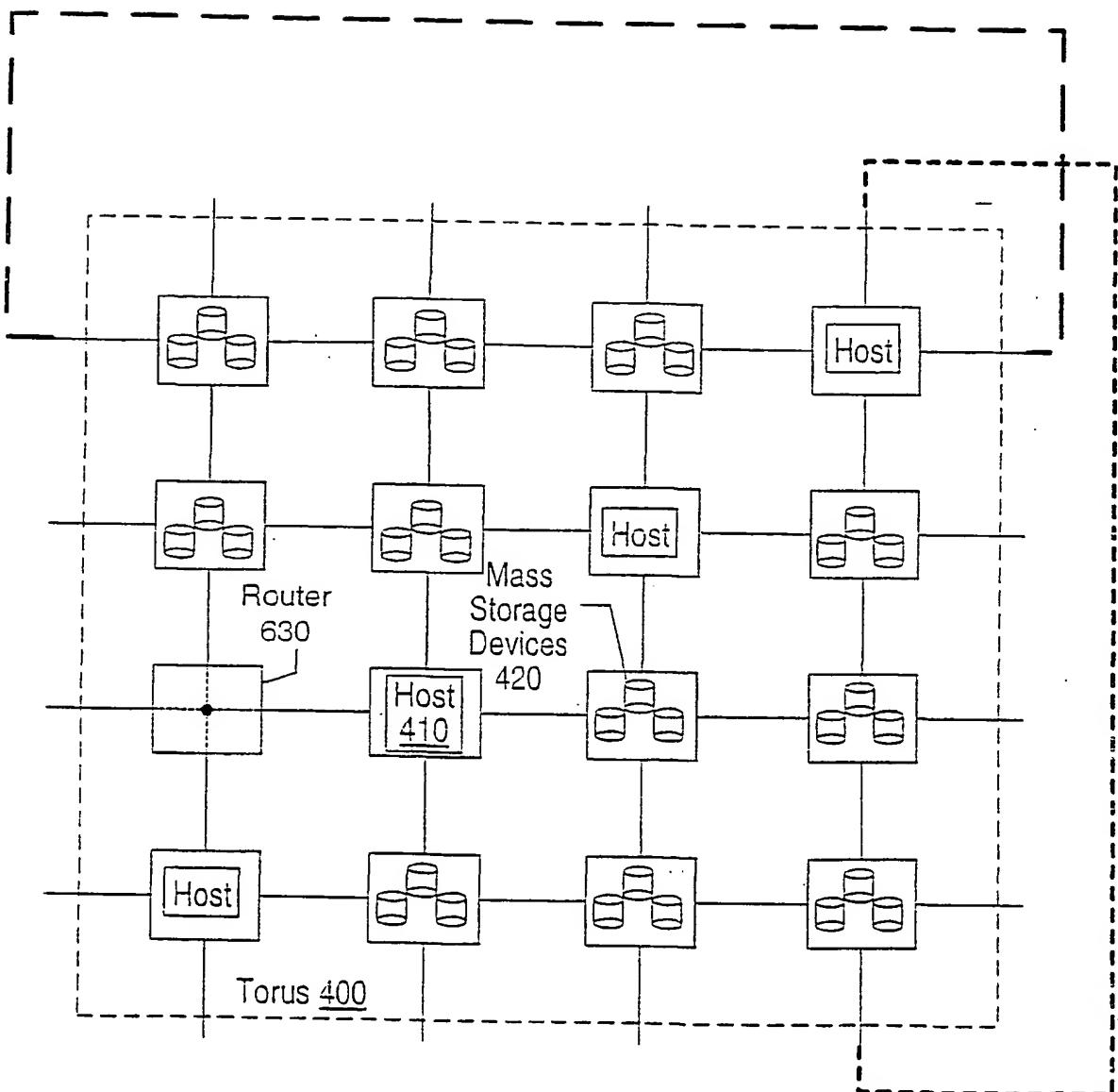


FIG. 3B

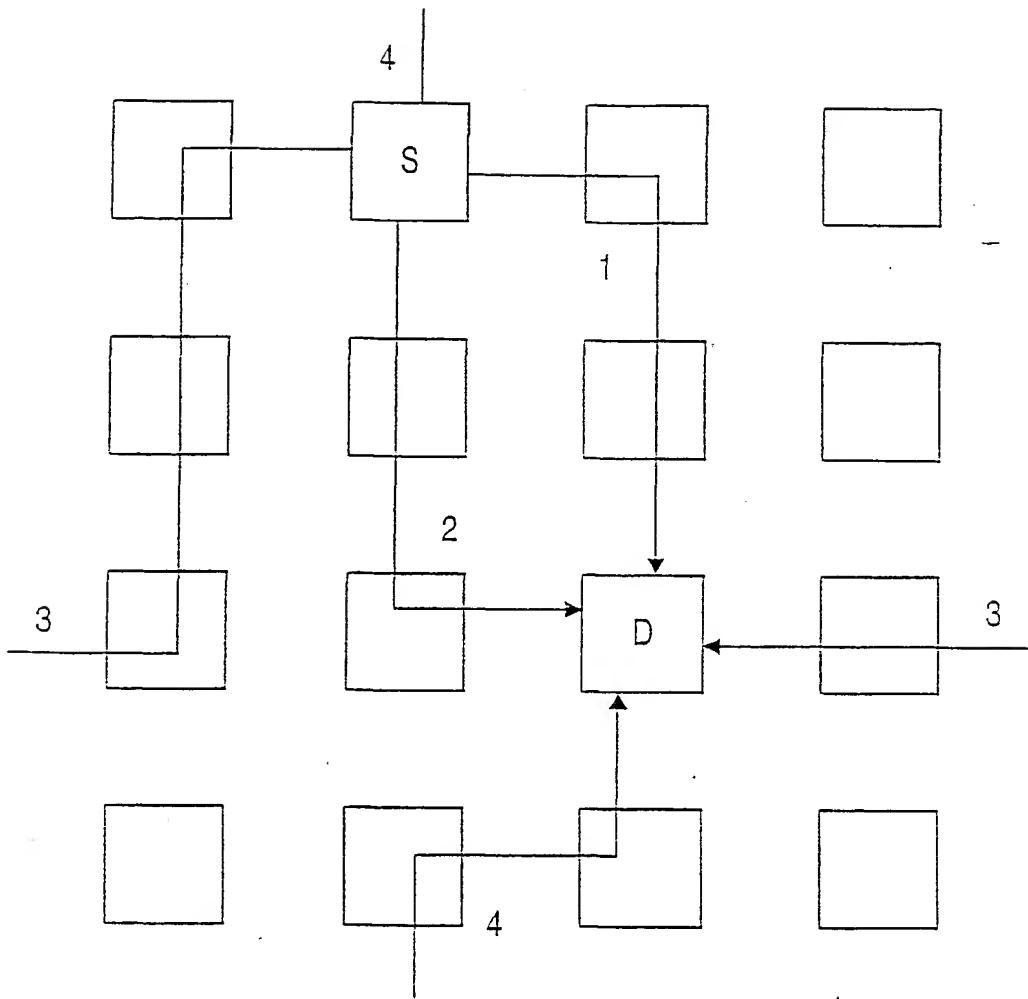


FIG. 4

6/12

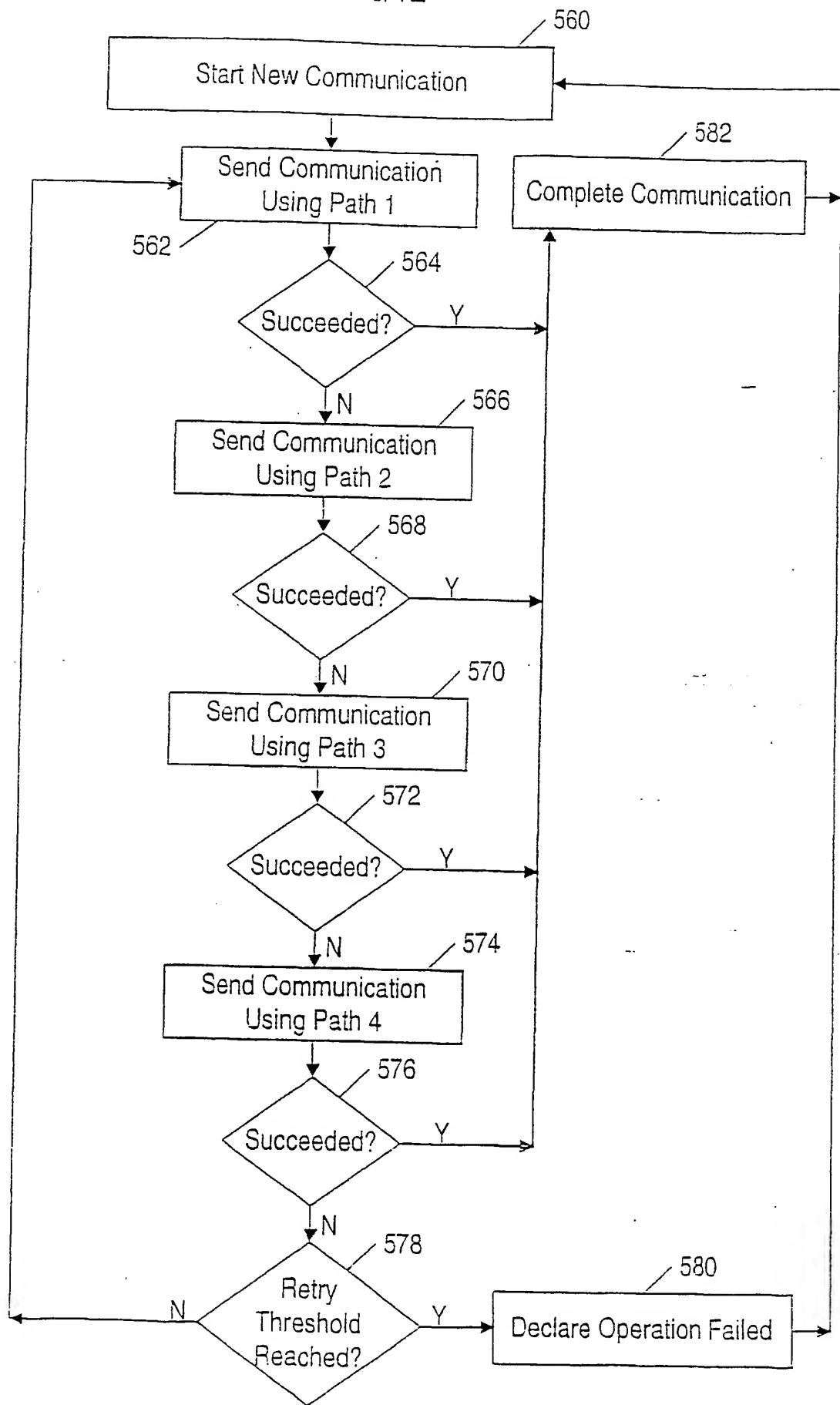


FIG. 5

7/12

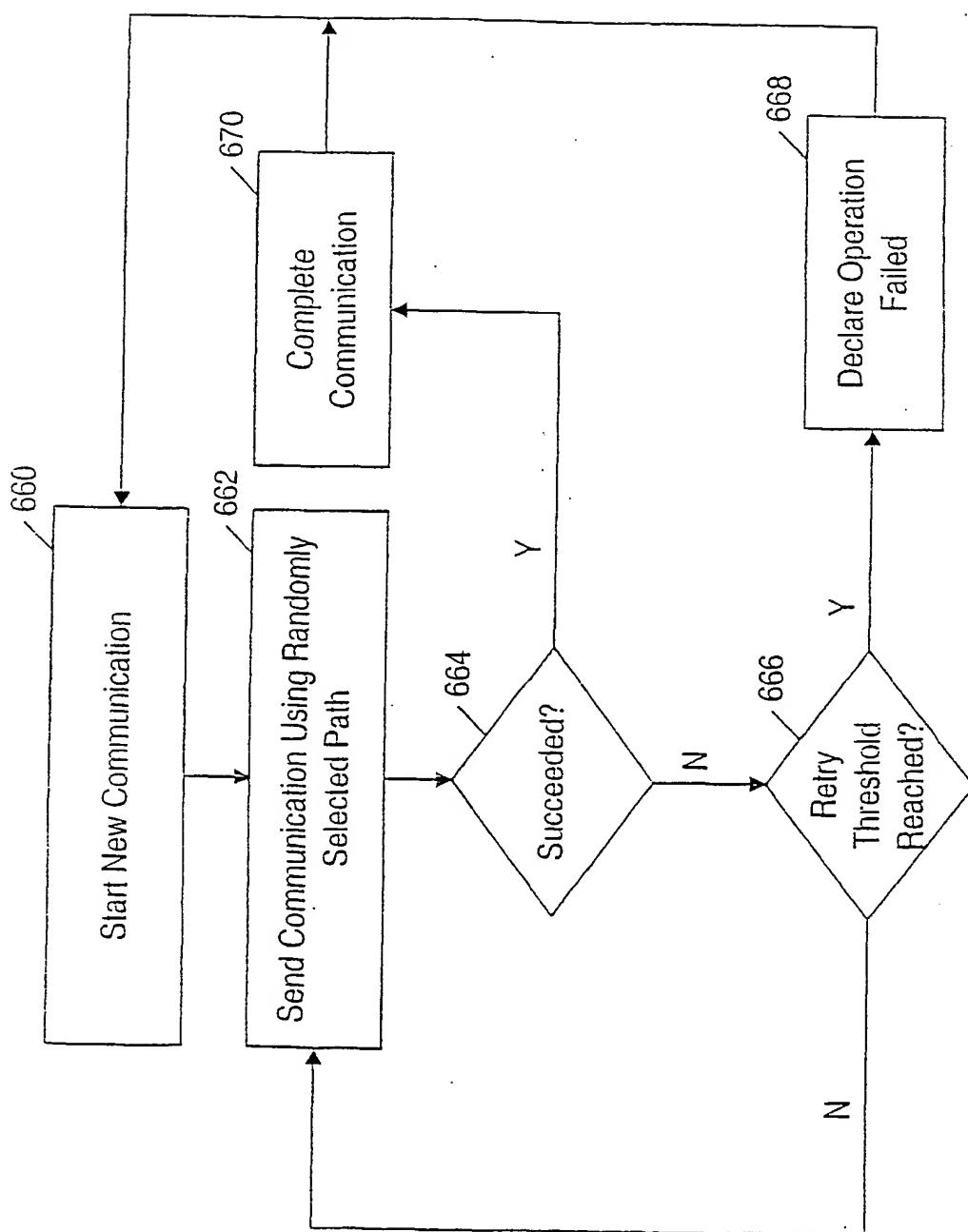


FIG. 6

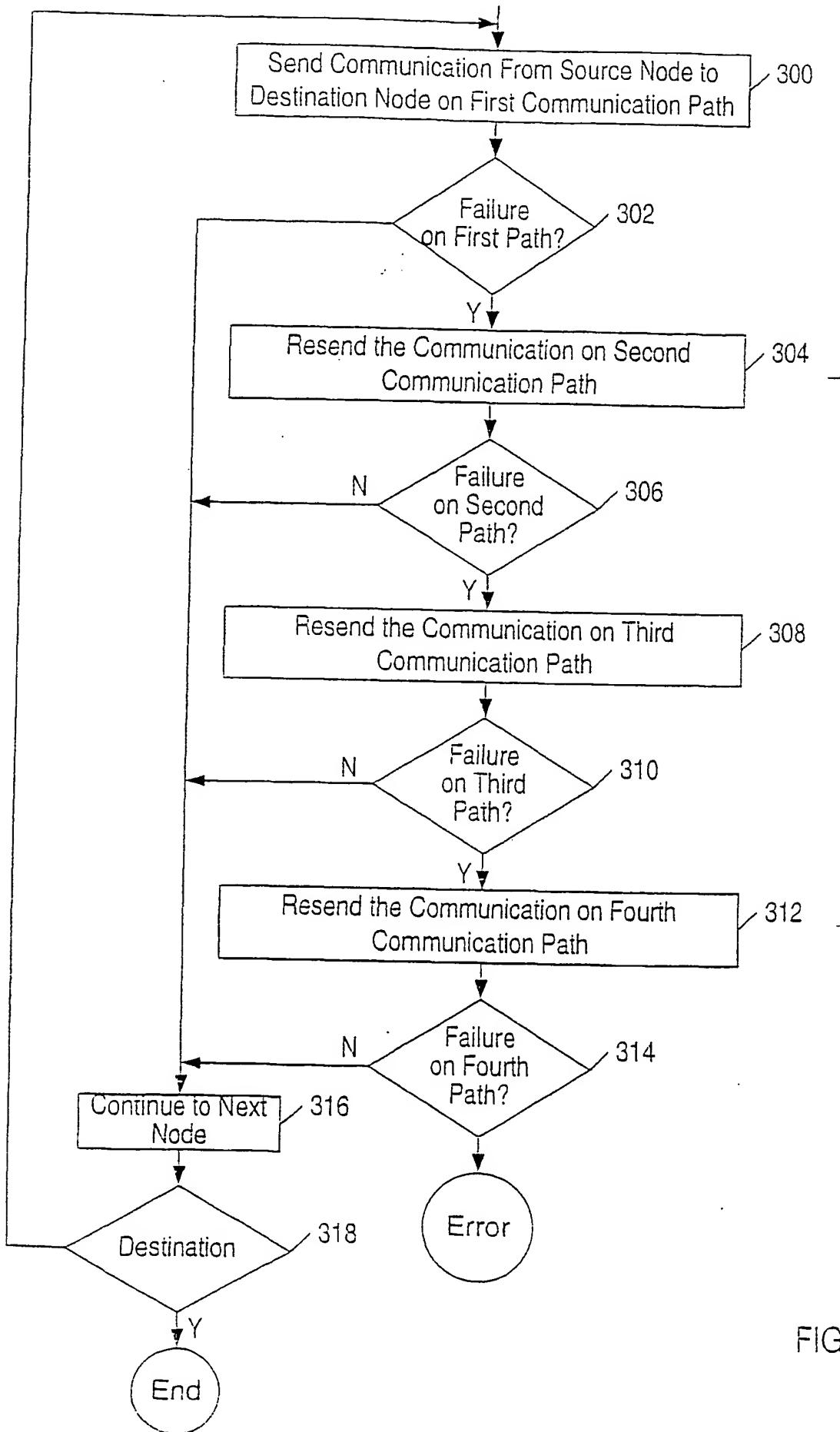


FIG. 7

9/12

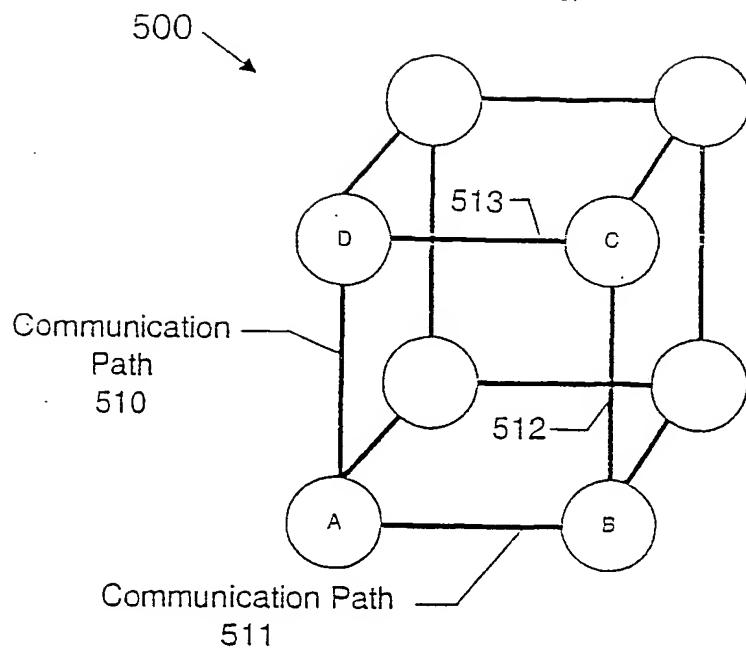


FIG. 8A

550

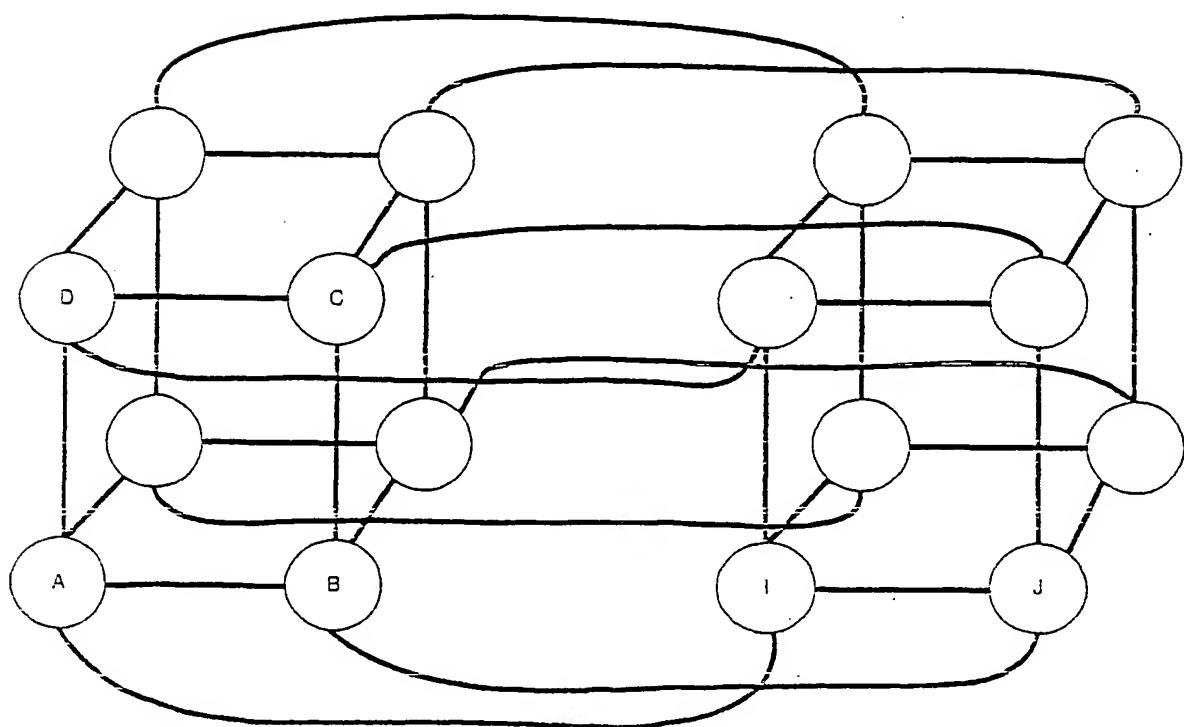


FIG. 8B

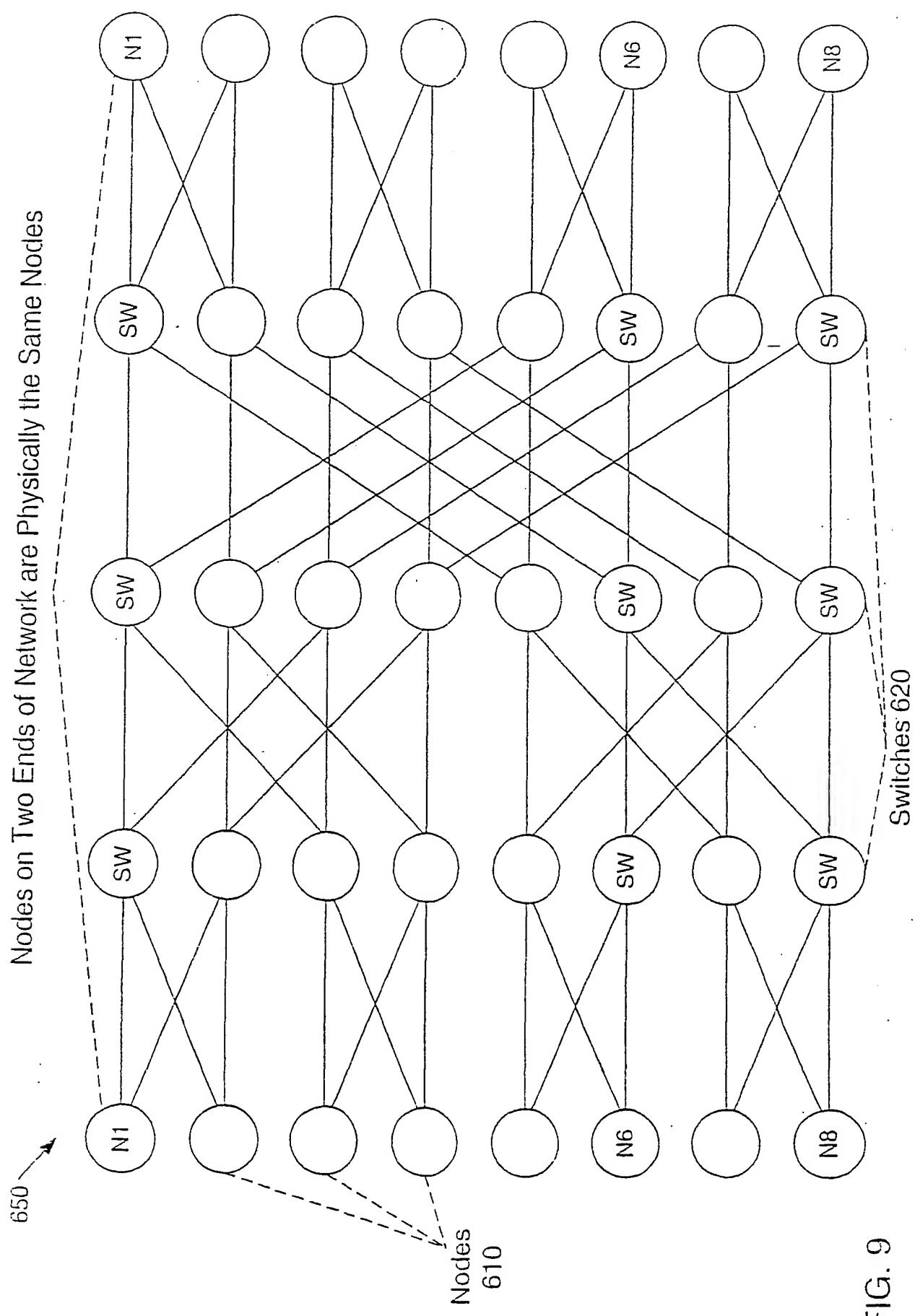


FIG. 9

11/12

700

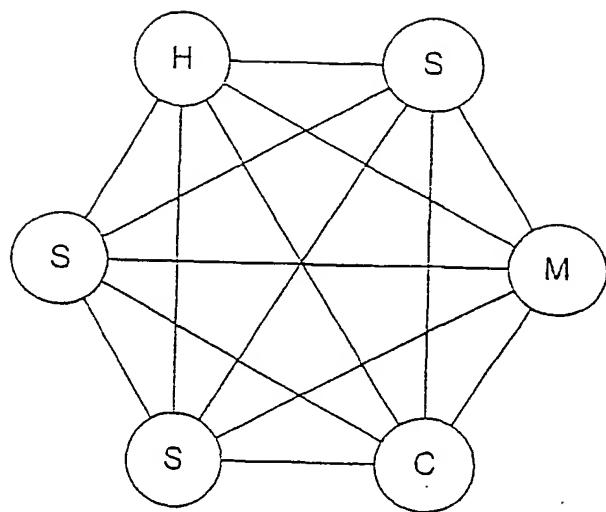


FIG. 10

800

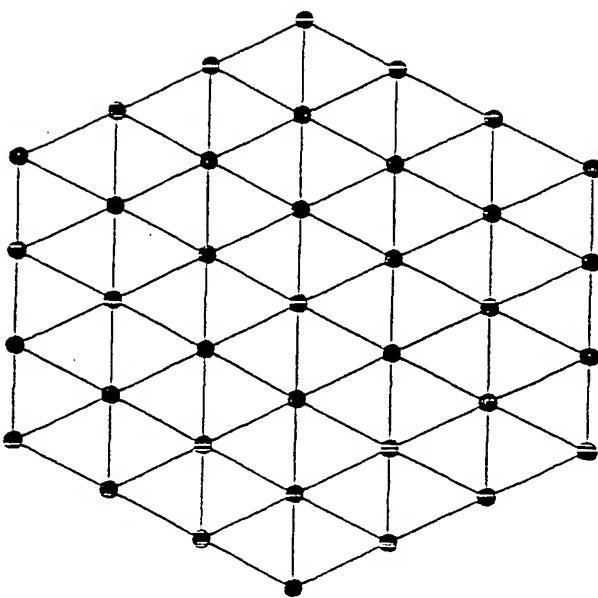


FIG. 11

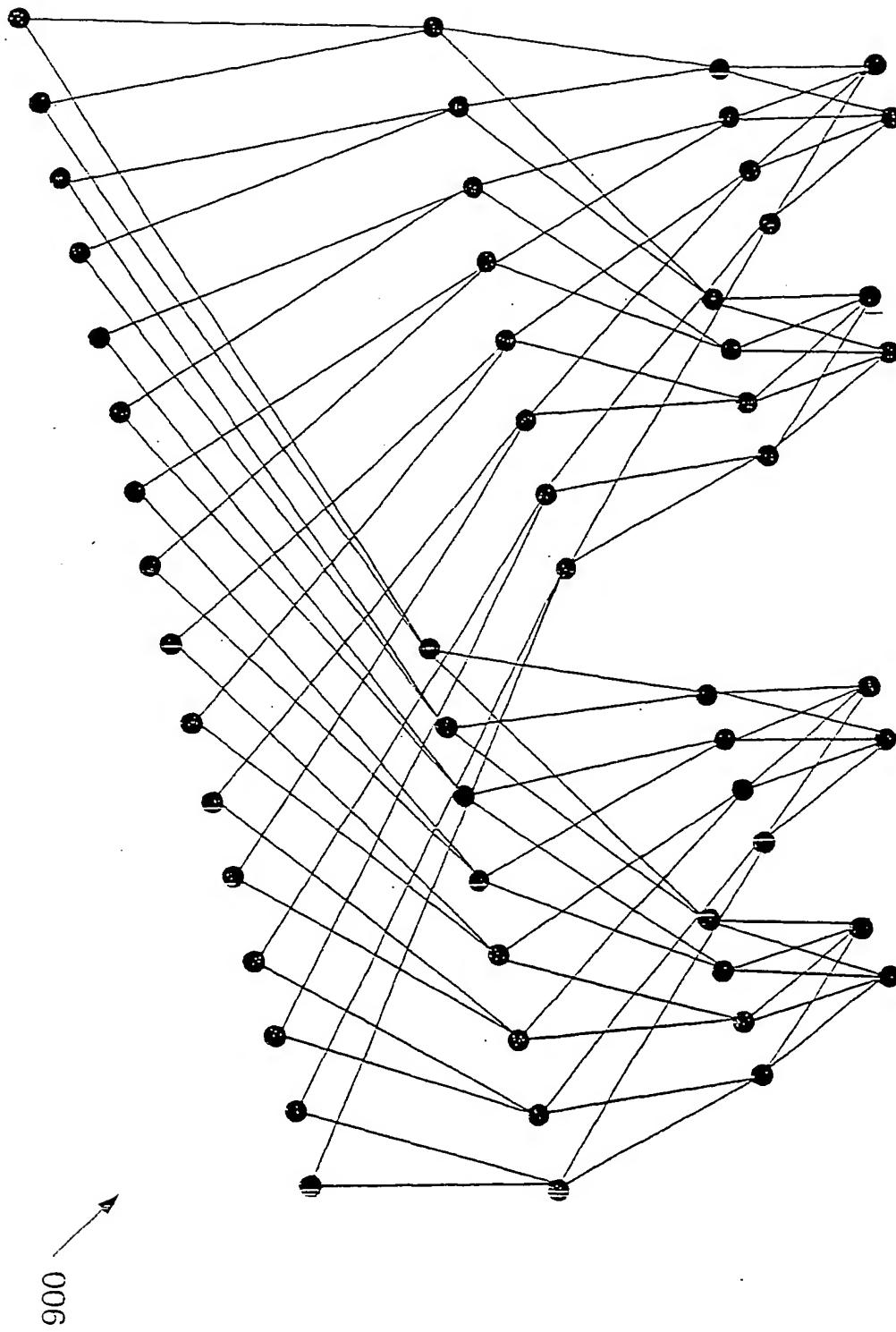


FIG. 12